

## REMARKS

In regard to the Examiner's Office Action of March 5, 2004, Applicant is supplying the following comments.

Applicant has reviewed the specification and has made some minor corrective amendments on the pages that are indicated.

On the question of anticipation which is involved in this case, it is noted that the filing date of the Bansal reference is August 15, 2002; while the filing date of Applicant's disclosure is October 30, 2001. Apparently, Examiner is relying on the Provisional filing date of August 15, 2001 in order to give Bansal credit for the earlier filing date of the Provisional.

A question that arises is `--- will Examiner provide Applicant with a copy of the Provisional application that was filed on August 15, 2001?

As noted in Applicant's specification the JAVA Database Connectivity application program interface assumes that the database being connected to is a "relational" database. These relational databases have flat table structures, or the tables consist solely of columns and rows.

However, the situation is quite considerably different when there is a "hierarchical" database involved. The hierarchical database allows tables to contain "embedded tables" or sub-tables. Such an organizational hierarchical concept is foreign to that of relational databases so that there is no support in the JDBC\_API to access the contents of sub-tables.

This prevents JAVA applications from using a standard database API (which the JAVA Database Connectivity is) from accessing data in a hierarchical database.

This presents a problem in that JAVA Database Connectivity is not normally useful to access data in a hierarchical database. Thus, the present invention fulfills the need to provide a mechanism that enables a JAVA application to use JDBC to access data in a hierarchical database.

Note that the hierarchical database --- is a database in which records are grouped in such a way that their relationships form a branching tree-like structure. In this regard, this type of tree-structured file system has folders which can be nested within other folders.

Here, each record may be the "parent" of one or more "child" records which may or may not have the same structure as the parent; a record can have no more than one parent. So conceptually, therefore, a hierarchical model can be, and usually is regarded as a tree.

The hierarchical organization is that like a tree which has branches into more specific units, each of which is "owned" by the higher level unit immediately above. These hierarchies are characteristic of several aspects of computing, because they provide organizational frameworks that can reflect logical links or relationships, between separate records, files or pieces of equipment.

For example, hierarchies are used in organizing related files on a disk, related records on a database, and related (interconnected) devices on a network.

Thus, by adding extra information to the present method of a JAVA application, it is then possible to access information from the hierarchical database.

This is done by determining the JDBC type, the JAVA class name, and the metadata values for the JAVA application in order to represent a given column in a result set (row) on the hierarchical database and then next, closing all embedded result sets for that particular column.

Attached as Appendix I, is a set of articles involving the characteristics of Hierarchical Databases.

Applicant is continuing claims 1-14 "as-is". Claim 15 has a spelling correction.

The Examiner has rejected Applicant's claims 1-15 under 35 USC 102(e) as being anticipated by the Bansal reference, U.S. Patent 2003/0120593A1. At this juncture, Applicant would transverse the Examiner's consideration as to anticipation of the claims of Applicant, as will be iterated below.

Clause 1(a). Regarding Applicant's clause 1(a) for --- determining a JDBC type for said JAVA application to represent a column in a result set on said hierarchical database --- here, the Examiner has cited page 21 of Bansal, paragraphs 0457-0459.

Here, it should be indicated that Bansal lists various mechanisms used to "connect" systems, one of those casually being JDBC. Applicant's disclosure is not about any "connection" mechanism, but rather involves accessing data in a hierarchical database using JAVA database activity (JDBC). It is noted that Bansal does not mention nor teach nor involve hierarchical databases.

Clause 1(b). Here, Applicant's clause indicates "determining a JAVA-class name for said JAVA application to represent said column ---. Here, it should be noted that there is no teaching in Bansal of any such functioning.

Clause 1(c). Here, Applicant shows --- creating a JAVA result set object to represent the data in said column . . .".

Here, Examiner cites page 3 of Bansal, paragraphs 0052-0057. We should note here that Bansal discusses the issues with that of qualifying various components that can be plugged into their architecture, but this connectivity has nothing to do with that of accessing a hierarchical database through use of JAVA Database Connectivity (JDBC). This has nothing to do with accessing a hierarchical database via JAVA Database Connectivity.

Clause 1(d). Here, Applicant states --- closing embedded result set objects . . .for said result set.

Here, Bansal cites page 25, paragraph 0533, where Bansal lists various types of content that can be accessed. However, none of the content mentioned is that of data in a hierarchical database. Further, Bansal makes no mention of "closing" anything.

Thus, it should be abundantly clear that the Bansal reference certainly does not teach the required aspects of Applicant's system.

Now, with regard to claim 2, the following comments are made:

Clause 2(a). Where Applicant states that --- determining if said column contains hierarchical data ---.

Here, Examiner has cited Bansal page 23, paragraph 0487-0489, where it will be noted that Bansal mentions casually, hierarchical databases, but not JAVA Database Connectivity. Quite contrarily, Applicant's system is about accessing of hierarchical data via JAVA Database Connectivity.

Clause 2(b). Here, Applicant involves --- setting said JDBC type to "JAVA.sql.types.other".

Here, Examiner cites Bansal page 27, paragraph 0581-0583, where Bansal merely talks about accessing data in a data warehouse via JDBC -- but does not include that of accessing "hierarchical data" via JAVA Database Connectivity.

Clause 2(c). Here, Applicant states --- determining in step (a) above, that said column does not contain hierarchical data--.

Here, Bansal does not mention any substance relating to this functionality.

Clause 2(d). Here, Applicant states -- handling said column in a normal fashion for non-hierarchical data--.

Here, Examiner cites page 23, paragraph 0487-0488, where nothing is taught about the handling of a column.

Now in regard to Applicant's claim 3, it will be seen that there is no teaching on page 23 of Bansal, paragraph 0487-0489 regarding the --- determining if said column contains hierarchical data.

Further, there is no teaching on page 27 of Bansal at paragraphs 0581-0583 regarding --- setting said JAVA class name type.

Further, there is no teaching in Bansal page 23, paragraph 0487-0488 regarding --- handling said column in a normal fashion for non-hierarchical data --.

Now, in regard to Applicant's claim 4, the Examiner has cited Bansal page 23, paragraph 0486-0488, as equivalent to Applicant's clause (b) --- creating a result set using said hierarchical data as contents. --- Here, it should be noted that these clauses of Bansal do not involve hierarchical data, as is focused in Applicant's system.

In regard to Applicant's claim 5(c), where Examiner has cited page 15 of Bansal, paragraph 0324 regarding Applicant's clause (c) --- adding said result set metadata object to a metadata collection ---.

Here, it is noted that the Bansal patent contains the term "metadata" which is used in the context of doing online searches. In Applicant's configuration, Applicant extends the existing metadata facilities which were built into JAVA Database Connectivity to determine if there is hierarchical data in the database.

The Bansal document makes no mention of extending the JDBC metadata facilities.

As a further note, the metadata in the Bansal disclosure, has nothing to do with the type of metadata in Applicant's configuration. In the Bansal disclosure, "metadata" is that part of an HTML page used to provide keywords or other descriptive information about the contents of the page, mainly to aid search pages.

Quite contrarily, in Applicant's configuration the "metadata" refers to the schema of the database which describes the layout of the tables in the database. Note that in each of these cases, the word "metadata" is a descriptive word, but the contexts involved are quite different and are not related.

Now, in regard to Applicant's claim 6, the Examiner has cited clause 6(b), by mentioning the Bansal reference at page 25, paragraph 0533, which involves --- closing each of said result set objects ---.

Again, it should be reiterated that the Bansal patent lists various types of content that can be accessed, but none of the content involved or mentioned is that of data in a hierarchical database, and further, Bansal makes no mention again of "closing" anything.

The Examiner has rejected claims 7-12 for the same reasons as that cited in claims 1-6. Here again, Applicant must traverse Examiner's considerations on the subject, since it is abundantly clear at this stage that Bansal does not teach or involve the use of JDBC for hierarchical database access. Further, none of the cited paragraphs in Bansal are on point in teaching or utilizing the action clauses of Applicant's claims.

It should now be suitably clear, that there is no way in which the Bansal reference can in any way provide the architectural configuration of Applicant's Fig. 1, nor can it teach the factors involved in Figs. 2-6, and especially does it not show or teach the flowchart activities involved in these figures.

It should be indicated that an application should be considered as a whole in its entirety, and the mere fact that one or two clauses of the claims may have some vague similarity to other technology, is no reason that the overall claim can be considered either anticipated, obvious, or invalidated by bits and pieces of other technology.

As a result, it is requested that Examiner take a more scrutinizing look at the various clauses of Applicant's claims and view them as a whole in their entirety, and appreciate the value thereof in accessing a hierarchical database with JAVA Database Connectivity.

It is now requested that Examiner appreciate the extent of Applicant's technology and provide a timely Notice of Allowance therefor.

Respectfully submitted,

By Alfred W. Kozak  
Alfred W. Kozak  
Reg. No. 24,265  
(858) 451-4615  
(949) 380-5822

---

Certificate of Mailing (37 CFR 1.8a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to the: MAIL STOP AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date:

May 24, 2004

Patti S. Preddy

Patti S. Preddy